

Amendments to the Specification:

Please amend the paragraph on page 10, lines 5-15, as follows:

Figure 4 is a UML diagram that illustrates a factory interface and the classes that are necessary to create factory objects in accordance with the present invention. Factory proxy instance version 1 including factory InvocationHandler instance 1 402, factory proxy instance version 2 including factory InvocationHandler instance 2 404, and factory proxy instance version 3 including factory InvocationHandler instance 3 406, all inherit from factory interface 400. Similarly to the description of **Figure 3**, only two classes, in addition to the interface class, are needed in order to create instance 402, instance 404, and instance 406. Proxy class 408, which is the same as Proxy class 320, and factory InvocationHandler class 410 are used to create factory proxy instance version 1 including factory InvocationHandler instance 1 402, factory proxy instance version 2 including factory InvocationHandler instance 2 404, and factory proxy instance version 3 including factory InvocationHandler instance 3 406.

Please amend the paragraph on page 10, lines 21-29, as follows:

Figure 5B illustrates a high level flow chart which depicts the process of an InvocationHandler in accordance with the present invention. Block 504 depicts a determination of whether or not any of the method's parameters are proxy instances. If a determination is made that at least one of the method's parameters are a proxy instance, the process passes to block 506 which [[505]] which ~~illustrates the parameters now matching the parameter format for the firmware API that is to be invoked.~~ Next, block 506 illustrates replacing each actual parameter that is a proxy instance with its associated underlying firmware API instance. Next, block 508 depicts replacing each formal parameter that is a proxy instance with its underlying firmware API object type. The process then passes to block 505 which illustrates the parameters now matching the parameter format for the firmware API that is to be invoked.

Please amend the paragraph on page 12, lines 9-15, as follows:

Block 610, then, depicts the factory proxy instance using Java reflection to create an interface InvocationHandler instance for this firmware version. Next, block 612 illustrates the factory proxy instance creating an interface proxy instance that implements the specified interface and includes a reference to the newly created interface InvocationHandler instance. Thus, the interface proxy instance may be thought ~~of as including the interface InvocationHandler instance.~~ The process then passes to block 614 which depicts the factory proxy instance returning the interface proxy instance which includes the interface InvocationHandler instance.

Please amend the paragraph on page 14, lines 11-18, as follows:

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media such as a floppy disc, a hard disk drive, a RAM, and CD-ROMs and transmission-type media such as digital and analog communications links.